Temporal reasoning

- Primitive entities are propositions with which we associate *temporal intervals*
- Temporal information may be
  - *Relative* or *metric*
  - *Interval* based or *time point* based
  - References to *absolute* time and *duration* of propositions

Simple Temporal Problems

- Variables are *time points*, $X_1, \ldots, X_n$
  - Represents the *beginning* or the *end* of an event
  - Domain is the *reals*
  - Special variable X_0 represents the "beginning of time" and is assigned the value 0 by convention.
- Constraints are *intervals* that constrain the distance between two variables
  - Constraint T_{ij} represented by the interval $[a_{ij}, b_{ij}]$ means
    $$a_{ij} \leq X_j - X_i \leq b_{ij}$$
  - Constraint T_{0i} constrains the domain of X_i
    $$a_{0i} \leq X_i - X_0 \leq b_{0i}$$

Simple Temporal Problems (contd.)

- An assignment $(X_1 = x_1, \ldots, X_n = x_n)$ is a *solution* if it satisfies all the constraints
- A problem is *consistent* if at least one solution exists

- A value v is a *feasible value* for a variable X_i, if there exists a solution in which $X_i = v$
- The set of all feasible values of a variable is called its *minimal domain*

- A problem is *decomposable* if every locally consistent assignment can be extended to a solution

Distance graph

- A temporal constraint T_{ij} is a pair of linear inequalities

$$X_j - X_i \leq b_{ij}$$
$$X_i - X_j \leq - a_{ij}$$

- Can be represented as a *distance* graph, in which an edge from X_i to X_j has weight w_{ij} representing constraint $X_j - X_i \leq w_{ij}$

- Example:
$10 \leq X1 \leq 20$
$30 \leq X2 - X1 \leq 40$
$10 \leq X2 - X3 \leq 20$
$40 \leq X4 - X3 \leq 50$
$60 \leq X4 \leq 70$

Shortest paths in the distance graph

- **Lemma:** $X_j - X_i \leq d_{ij}$, where d_{ij} is the shortest distance from X_i to X_j in the distance graph

Consistency of simple temporal problems

- **Theorem:** A simple temporal problem is consistent if and only if its distance graph has no negative cycles
- **Corollary:** Given a consistent simple temporal problem, the following are consistent solutions
  - $\{X_1 = d_{01}, \ldots, X_n = d_{0n}\}$
  - $\{X_1 = -d_{10}, \ldots, X_n = -d_{n0}\}$

Decomposability

- **Theorem:** Any consistent simple temporal problem is decomposable relative to the constraints in its distance graph
- **Corollary:** The set of feasible values for a variable X_i is $[-d_{i0}, d_{0i}]$

Shortest path problems

- d_{0i} can be computed using a *single source shortest path* algorithm applied to the distance graph
- d_{i0} can be computed using a *single destination shortest path* algorithms applied to the distance graph
  - Can be reduced to a single source shortest path computation

Shortest path algorithms

- Algorithms maintain
  - d_i : current best *estimate* of shortest path from source
  - π_i : *predecessor* node in the shortest path

function *initialize-single-source*(G, s)

for each vertex $v \in$ *vertices*(G) **do**

$d_v = \infty$

$\pi_v = nil$

endfor

$d_s = 0$

end *initialize-single-source*

Relaxation

function *relax*(u, v)

  if $d_v > d_u + w_{uv}$ **then**

    $d_v = d_u + w_{uv}$

    $\pi_v = u$

    return *true*

  else

    return *false*

  endif

end *relax*

Bellman-Ford algorithm

function *bellman-ford*(G, s)

initialize-single-source(G, s)

for i = 1 **to** |*vertices*(G) – 1| **do**

changes = *false*

for each edge $(u, v) \in$ *edges*(G) **do**

if *relax*(u, v) **then** *changes* = *true* **endif**

endfor

if *changes* == *false* **then return** *true*

endfor

for each edge $(u, v) \in$ *edges*(G) **do**

if $d_v > d_u + w_{uv}$ **then return** *false* **endif**

endfor

return *true*

end *bellman-ford*